AN INTRODUCTION TO ATLAS PIXEL MODULE CALIBRATION PROCEDURES

John Richardson

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA

1. INTRODUCTION

This document is intended to serve as an overview of the procedures required in performing Pixel module tests, including two types of calibration. The command structures for the Module Controller Chip (MCC) and Front-End (FE) chips are described along with the format of the upstream data. The sequences of commands which would be executed during a one-dimensional threshold scan and a Time-Over-Threshold calibration scan are also described. The commands and procedures described here are based upon the first Deep Submicron realisations of the MCC (MCC-I1) and FE (FE-I1), the details of which are subject to change in future versions.

2. MCC COMMAND STRUCTURE

MCC-I1 has a single serial command interface, therefore all means of communicating with the MCC are by means of a data line (DCI) along with the 40MHz master clock (XCK).

Each MCC command has 5 sections thus:

<HEADER><FIELD 2><FIELD 3><FIELD 4><FIELD 5>

Field 2 is always 4 bits, field 3 is 4 bits and field 5 is variable. There are three basic types of command; Trigger, Fast and Slow:

Type	Name	Header	Field 2	Field 3	Field 4	Field 5	F5 BITS
Trigger	LV1	11101					N/A
Fast	BCR	10110	0001				N/A
Fast	ECR	10110	0010				N/A
Fast	CAL	10110	0100				N/A
Fast	SYNC	10110	1000				N/A
Slow	WriteReg	10110	1011	0000	Address	Data	16
Slow	ReadReg	10110	1011	0001	Address	-	16
Slow	WriteFifo	10110	1011	0010	-	Data	21
Slow	ReadFifo	10110	1011	0011	Address	-	21
Slow	WriteFE	10110	1011	0100	-	FE cmd	Variable
Slow	WrRdFE	10110	1011	0101	-	FE cmd	Variable
Slow	WrReceiver	10110	1011	0110	-	Data	Variable
Slow	EnDataTake	10110	1011	1000	-		0
Slow	ResetMCC	10110	1011	1001	-		0
Slow	ResetFE	10110	1011	1010	-	SyncW	4

Fields labelled “-“ should be filled with 4 zeros in the case of field 4 and with zeros for the number of bits given in the last column for field 5. Blank fields are irrelevant to the command in question.

LV1: Issues n contiguous triggers to the FE chips (n stored in LV1 register)

BCR: Resets the bunch-counter to 0 in the MCC

ECR: Resets the event counter in the MCC

CAL: Issues a calibration strobe to the FE chips, the attributes of which are stored in the CAL and CNT registers

SYNC: Generates a SYNC pulse for the FE chips

WriteReg: Write to an MCC register. MCC-I1 has 8 internal registers, those being:

Name	Address	Content
CSR	0000	Overall control and status register
LV1	0001	LV1 trigger info. (# contiguous triggers, LV1 count)
FEEN	0010	Front-end enable (1 bit per FE)
WFE	0011	Warning from FE chips (R/O)
WMCC	0100	Warning from MCC receiver (R/O)
CNT	0101	Control (CCNT) and data length (DCNT) for FE slow control commands (also CAL strobe length)
CAL	0110	CAL strobe attributes (fine delay, delay range)
PEF	0111	Pending event FIFO register (R/O)

ReadReg: Reads the contents of an MCC register, (field 5 should be filled with zeros)

WriteFifo: Writes a word into any receiver FIFOs enabled in FEEN (diagnostic)

ReadFifo: Reads the content of a receiver FIFO (diagnostic)

WriteFE: Issues a slow control command to the FE chips. Field 5 in this case is the FE command itself. Note that the FE command protocol is at 5MHz therefore each FE command bit is written 8 times in the (40MHz) MCC command structure. The FE command is composed of a control part (with CCNT 5MHz bits) and an optional data part (with DCNT 5MHz bits), (see later). CCNT and DCNT must be written to the CNT register prior to the execution of FE commands. CCNT is quantised in units of 8 and for FE-I1 is always = 32 (since the control register in FE-I1 is 25 bits long). DCNT can be anything from 0 to 2880 thus the longest MCC command possible in XCK units is
5 + 4 + 4 + 4 + 8*(32 + 2880) = 23313 bits.

WrRdFE: Also issues a slow control command to the FE chips. This time however the MCC anticipates return data from the FE and passes it through to the serial data output. The length of the FE return data string in XCK units is 8*(32+DCNT) for FE-I1 to which the MCC adds a 5-bit header.

WrReceiver: Writes data to enabled reciever channels to emulate FE slow control data (diagnostic).

EnDataTake: Arms the MCC for subsequent LV1 command reception, must be issued before normal data taking. Sending another slow command will take the MCC out of this state.

ResetMCC: Resets all of the internal registers, FIFO pointers etc. in the MCC.

ResetFE: Propogates a SYNC pulse to the FE chips of width SyncW in BCO units.

<u>3. MCC REGISTER FORMATS</u>

The register contents in the MCC are defined thus:

CSR[12:0] Control and Status Register:
CSR[12:8] Status Register {LV1FlagEVB, LV1FlagRec, WngSlow, WngFast, WngLV1}
CSR[6:4] Control Register {PLBK, OPAT, TOT}
CSR[3:0] Control Register {ECHK, EREP, OM[1:0]}

LV1[11:0] LV1 Register:
LV1[11:8] LV1_cont: Number of contiguous Level 1s to be issued to the FE chips
LV1[7:0] LV1IdCnt: Level 1 ID counter (mapped from the TTC, not writable by Write_Reg)

FEEN[15:0] Front End Enable, (one bit per FE, LSB = chip 0)

WFE[15:0] Warning from FE (Read/Only)

WMCC[15:0] Warning from MCC Receiver (R/O)

CNT[15:0] Counter Register:
CNT[15:3] Data bits transmitted to the FE chips (DCNT)
CNT[2:0] Control bits transmitted to the FE chips (bits x 8) (CCNT)
CNT[15:0] Signal width of the Delay pulse when not issuing commands to FE chips

CAL[10:0] Calibration Pulse Delay Settings:
CAL[10] Static power On/Off
CAL[9:6] Fine delay range
CAL[5:0] Fine delay

PEF[15:0] Pending Event Fifo Register:
PEF[15:12] Skipped events
PEF[11:8] L1ID
PEF[7:0] BCID

PEF is mapped to the PendingLV1Fifo.
Read and write pointers are incremented by the corresponding Read_Reg, Write_Reg commands. This fifo can also be read and written by the TTC.
The Fifo is written only if in Run Mode.

4. FE COMMAND STRUCTURE

The front-end (FE) commands are enveloped within the WriteFE and WrRdFE MCC commands as field 5. The FE control protocol is at 5MHz, therefore each bit must be written 8 times within field 5. The FE command is split into 2 sections, the command section and data section. Most FE commands only use the command section, however whenever FE registers are lo aded the data section is used to carry the appropriate data.
There are three main registers in the FE chip, Command, Global and Pixel. When a command is issued, the command section is loaded into the Command register. The last five bits of the command section contain the geographical address (GA) of the FE chip.

Name	Command	DCNT
NULL	0x00000	0
SOFT RESET	0x00001	0
CLOCK GLOBAL	0x00002	202
WRITE GLOBAL	0x00004	0
READ GLOBAL	0x00008	0
CLOCK PIXEL	0x00010	320*n (n = 0 to 9)
WRITE HITBUS	0x00020	0
WRITE SELECT	0x00040	0
WRITE KILL	0x00080	0
WRITE MASK	0x00100	0
WRITE TDAC0	0x00200	0
WRITE TDAC1	0x00400	0
WRITE TDAC2	0x00800	0
WRITE TDAC3	0x01000	0
WRITE TDAC4	0x02000	0
WRITE FDAC0	0x04000	0
WRITE FDAC1	0x08000	0
WRITE FDAC2	0x10000	0
WRITE FDAC3	0x20000	0
WRITE FDAC4	0x40000	0
READ PIXEL	0x80000	0

Each FE chip within a module has a unique 4-bit GA which is programmed using 4 input pads, (the layout of the flex hyrid circuit provides the appropriate GA as long as all 4 bons are completed on each chip). The GA is independent to the FE ID according to MCC, (i.e. the FEEN bit associated with a particular FE, which corresponds to the MCC receiver channel that the FE is hard wired to, has no relation to the GA). However the intention is that modules should be configured in such a way that they are actually the same. The fifth bit in the GA command field is a broadcast bit to which all FEs will respond. Upon receipt of a command, if the GA is recognised (or broadcast bit set) the

command will be decoded. If the command is a CLOCK PIXEL or CLOCK GLOBAL, then the subsequent data contained within the data field will be clocked into the register in question at 5MHz.
The MCC must be made aware of the lengths of the command section and data section of the FE command by setting the CNT register to the appropriate values. The command section field of this register (CCNT) is quantised in multiples of 8, therefore for FE-I, although the command register is only 25 bits long, the value 32 must be used. Then he redundant 7 'bits' at the beginning are just padded with zeros and the true command field is sent within the last 25 bits. In this way the extra leading zeros are merely clocked off the end of the command register. The data-section field (DCNT) is set to the exact number of data bits present in the FE command.

NULL: This command must be issued after every 'data-less' command (i.e. all except CLOCK GLOBAL and CLOCK PIXEL) in order to activate it.

SOFT RESET: Performs a reset of the digital section of the FE chip.

CLOCK GLOBAL: Clocks the subsequent 202 bits of data in the data field into the Global register.

WRITE GLOBAL: Latches the Global register contents.

READ GLOBAL: Copies the contents of the Global latches back into the Global register.

CLOCK PIXEL: Clock the subsequent data into the pixel register. The number of bits is a multiple of 320, (320 being the number of pixels in each column pair). Generally all 9 column pairs would be operational (and therefore enabled) giving a DCNT of 2880.

WRITE HITBUS: Copy the contents of the pixel register into the Hitbus-enable control latches (one present in each pixel, 1=On).

WRITE SELECT: Copy the contents of the pixel register into the Calibration-injection-enable control latches (one present in each pixel, 1=On).

WRITE KILL: Copy the contents of the pixel register into the Preamplifier-kill control latches (one present in each pixel, 1=Kill).

WRITE MASK: Copy the contents of the pixel register into the Digital-readout-enable control latches (one present in each pixel, 1=On).

WRITE TDAC0: Copy the contents of the pixel register into the LSB of the Threshold-trim-DAC.

WRITE TDACn: Copy the contents of the pixel register into the nth bit of the

Threshold-trim-DAC.

WRITE FDAC0: Copy the contents of the pixel register into the LSB of the Feedback-trim-DAC.

WRITE FDACn: Copy the contents of the pixel register into the nth bit of the Feedback-trim-DAC.

READ PIXEL: Turn off the 'Shift' signal which is used in the Pixel control bit readback process.

5. FE REGISTER FORMATS

5.1 GLOBAL REGISTER:

This register is composed of 202 bits, the MSB of which would be clocked into the chip first:

Reg(0:7) = Latency: The delay in BCO units between instances of hits and accepted triggers (LV1s) to which they would be associated and thus read out.

Reg(8:11)* = Self Trigger Width: The number of internally generated L1 signal in self-trigger mode.

Reg(12)* = EnableSelfTrigger: Activates the self-triggering circuitry which is designed to automatically generate internal LV1 signals when the hitbus fires.

Reg(13:16) = DO-MUX: Select the source of data which is transmitted on the DO pin of the FE chip:

0 = RegClkIn, buffered CCK, gated for entry to Glo bal register
1= HitBus
2 = SoftResetB, inverted SoftReset signal
3 = SerData, data from digital readout of chip
4 = MonSel, chip select status
5 = PixClk, CCK, input to 2-phase clock for Pixel register
6 = digital ground
7 = digital VDD
8 = SerData, data from digital readout of chip: **Use this value for normal event data taking**
9 = SerOutMain, output data from Command register
10 = ReadFIFO, output from readout controller to read FIFO
11 = SerPix, output data from Pixel register: **Use this value when perfroming readback checks on the Pixel register**
12 = Accept, internal LVL1 signal for EOC control
13 = ROCK, internal readout clock for data transfer from EOC

14 = SerOutReg, output data from the Global register after place and route block (first 23 bits).
15 = GlobalOut, output data from Global register: **Use this value when performing readback checks on the Global register**

Reg(17:20)* = Select_MonHit.

Reg(21) = TSI/TSC Enable: Enable bit for the buffers on the output of the Grey Counter. When this bit is cleared, the TSI and TSC distribution to the column pair and EOC buffers is suppressed in order to reduce digital power consumption to a minimum.

Reg(22)* = Spare

Reg(23:35)* = MonLeak ADC circuitry.

Reg(36:41)* = CapMeasure Circuitry.

Reg(42)* = EnableCapTest.

Reg(43)* = EnableAnalogOut.

Reg(44:45)* = TestPixelMUX.

Reg(46)* = EnableVCalMeas.

Reg(47)* = EnableLeakMeas.

Reg(48)* = EnableBufferBoost.

Reg(49) = EnableCol8: set to enable column pair 8 operation.

Reg(50)* = TestDAC for IVDD2 DAC.

Reg(51-58) = IVDD2 DAC setting

Reg(59-66) = ID DAC setting

Reg(67)* = TestDAC for ID DAC.

Reg(68) = EnableCol7: set to enable column pair 7 operation

Reg(69)* = TestDAC for IP2 DAC.

Reg(70-77) = IP2 DAC setting.

Reg(78-85) = IP DAC setting.

Reg(86)* = TestDAC for IP DAC.

Reg(87) = EnableCol6: set to enable column pair 6 operation.

Reg(88)* = TestDAC for ITrimTh DAC.

Reg(89-96) = ITrimTh DAC setting.

Reg(97-104) = IF DAC setting

Reg(105)* = TestDAC for IF DAC.

Reg(106) = EnableCol5: set to enable column pair 5 operation

Reg(107)* = TestDAC for ITrimIf DAC.

Reg(108-115) = ITrimIf DAC setting

Reg(116-124) = VCal DAC setting – NOTE this is a special 9-bit DAC

Reg(125)* = TestDAC for VCal DAC.

Reg(126) = EnableCol4: set to enable column pair 4 operation

Reg(127) = High calibration cap selection. If low, injection uses the low injection cap only. If high, both the low and high value injection caps are used.

Reg(128)* = EnableExtInj.

Reg(129)* = Test Analog Reference.

Reg(130-131)* = EOC MUX control.

Reg(132-133) = CEU Clock Control: The CEU clock controls the speed of operation of the column pair readout:
0 = turn CEU clock off to save power
1 = CEU clock = 10MHz (5MHz column transfer rate)
2 = CEU clock = 20MHz (10MHz column transfer rate)
3 = CEU clock = 40MHz (20MHz column transfer rate)

Reg(134) = Enable digital injection: enable for passing the strobe signal directly into the pixel hit logic for artificial ‘hit’ creation.

Reg(135) = EnableCol3: set to enable column pair 3 operation.

Reg(136)* = TestDAC for ITH1 DAC.

Reg(137-144) = ITH1 DAC setting.

Reg(145-152) = ITH2 DAC setting.

Reg(153)* = TestDAC for ITH2 DAC.

Reg(154) = EnableCol2: set to enable column pair 2 operation.

Reg(155)* = TestDAC for IL DAC.

Reg(156-163) = IL DAC setting.

Reg(164-171) = IL2 DAC setting.

Reg(172)* = TestDAC for IL2 DAC.

Reg(173) = EnableCol1: set to enable column pair 1 operation

Reg(174-181) = THRMIN, minimum threshold register for CEU TOT processor

Reg(182-189) = THRDUB, minimum threshold register for CEU hit doubling

Reg(190-191) = ReadMode:
0 = send hits to the EOC in the usual way
1 = apply THRMIN threshold to TOT value for each hit and only write those hits to the EOC which have TOT greater than THRMIN.
2 = for hits with TOT less than or equal to THRDUB, the hit will be sent twice to the EOC. One hit has the original LE timing information, and one hit has the LE timing information for the previous beam crossing. The CEU generates one cycle of wait state for the column readout in order to write twice to the EOC.
3 = perform both of the above operations

Reg(192) = EnableCol0: set to enable column pair 0 operation

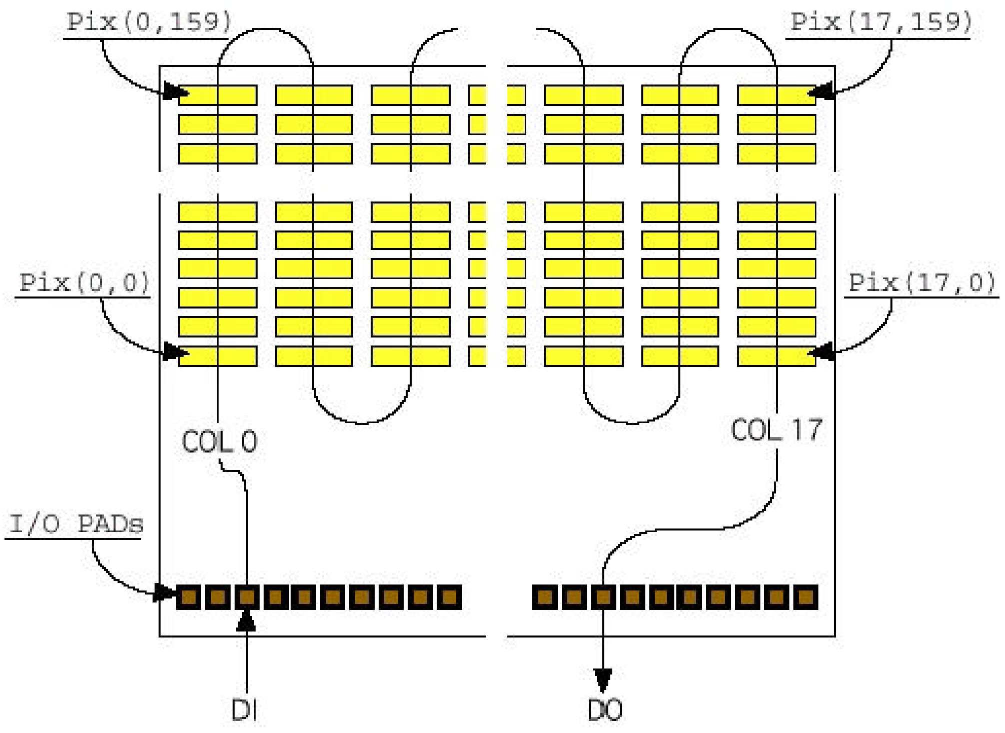
Reg(193)* = TestDAC for SPARE DAC.

Reg(194-201)* = SPARE DAC setting.

Fields marked * will not normally be utilised when operating full modules with RODs and should therefore be set to zero.

5.2 PIXEL REGISTER

The pixel register has a total of 2880 bits, visiting every pixel on the FE chip. It is divided up into column pairs (CP) and may be 'short-circuited' at each CP by disabling the CP in question. There are 9 column pairs in total with 320 pixels each. Therefore the number of data bits which should be loaded during a CLOCK PIXEL command, (and the value of DCNT), should be equal to the number of enabled CPs multiplied by 320. The register is arranged on the chip in a pettern which snakes up and down the column pairs, such that (with all CPs enabled) the pixel with row number 0 and column number 17 [17,0] will be the first to be clocked in and [0,0] the last:



The purpose of the pixel register is to deliver local control information to the pixel cells. Each pixel cell incorporates 14 bits of local control; there are individual bits to enable the pixel for readout (ENABLE), select the pixel for calibration injection (SELECT), select the pixel to participate in the discriminator output global-OR (HITBUS) and to kill the preamplifier by disabling its bias current (KILL). The remaining 10 bits constitute two 5-bit trim DACs for local threshold tuning (TDAC) and feedback current, (TOT-scale therefore), tuning (FDAC). For each bit there exists an FE command to provide the load which causes the current content of the pixel register to be latched. There also exists the ability to readback the content of individual control latches into the pixel register. This is not available as a single command, rather a compound set of commands must be contructed which involves setting several command bits simultaneously. First the READ PIXEL bit is set (MSB in the command register), then the LOAD bit corresponding the control bit in question is also enabled. Next, while holding these two bits on a double CCK pulse is issued by turning on the CLOCK PIXEL bit accompanied by a 2 bit data field and then turning it off again. Then the LOAD bit is disabled and finally the READ PIXEL bit is turned off. A full CLOCK PIXEL command would then be issued with a 2880-bit data field in order to strobe the data off the chip. This procedure is particularly convoluted when talking to an FE chip through an MCC since the DCNT field in the CNT register must be altered in the middle of the sequence, for the small 2-bit CLOCK PIXEL command.

6. MCC TEST PROCEDURES

In order to verify the basic functionality of the MCC one may perform write-read tests on those register fields which are r/w enabled. One may also write random data to the FE input FIFOs and read back the data directly along with writing dummy FE data to the the FE receivers as a means of testing the event building machinery.

6.1 MCC REGISTER TEST

The basic method is to write a predetermined data pattern to the register in question and read it back to compare with expectation. It's good practice to test each bit with a 1 and with a 0. In TurboDAQ e.g. the technique used involves writing 0xAAAAs and then 0x5555s seperately. The available r/w enabled fields are as follows:
CSR[12:8], CSR[6:0], LV1[11:8], CAL[10:0] and [15:0] for the other registers.
The following is an example test sequence in which a single register field is tested with 0x5555:

ResetMCC;
10110101110010000
WriteReg; register = FEEN, data = 0x5555
10110101100000010010101010101010101
ReadReg;
10110101100010010000000000000000

The return data stream should be;
111010101010101010101
the first five bits representing the MCC to ROD protocol header field followed by the 16 bits of data.

6.2 MCC FIFO TEST

The input FIFO has 128 locations therefore in order to test the FIFO in its entirety, one would make 128 consecutive writes followed by 128 reads. The FEEN register must be written to, in order to select the FIFO(s) taking part in the test. When reading the FIFO(s) the FIFO ID is in the command since only a single FIFO can be read at a time. N.B. in MCC-I1 only the upper 21 bits (of 27) are available for r/w tests. In this example FIFO #5 is tested with the 0x155555 pattern:

ResetMCC;
10110101110010000
WriteReg; register = FEEN, data = 0x0020 (FIFO 5)
1011010110000001000000000000100000
128 X WriteFifo; data=0x155555
10110101100100000101010101010101010101
10110101100100000101010101010101010101
.
.
.
10110101100100000101010101010101010101
128 X ReadFifo; address = 5
10110101100110101000000000000000000000
.
.
etc.

For each ReadFifo the return stream in this case should be:
11101101010101010101010101

6.3 MCC EVENT BUILD TEST

For this type of test there is complete freedom to define what the 'events' look like and to create them using dummy FE data, which is written to the receiver channels using the WrReceiver command. Serial data streams are quantised in bytes and the number of bytes must be declared in the CNT register. A simple test might involve writing one hit and one end-of-event (EoE) word to the receiver for FE index 2:

ResetMCC;
10110101110010000
WriteReg; register = CSR, data = 0x5C (Enable event playback etc.)
1011010110000000000000000001011100
WriteReg; register = LV1, data = 0x0000 (Request a single level-1)
1011010110000000100000000000000000
WriteReg; register = FEEN, data = 0x0004 (FIFO 2)
1011010110000001000000000000000100
WriteReg; register = CNT, data = 0x0004 (Defines number of bytes of data for subsequent WrReceiver command)
1011010110000010100000000000000100
WriteReceiver; with dummy FE hit data (BCID 2, column 1, row 1, TOT 15) padded out to 32 bits
10110101101100000100100000000100001000011110000000
WriteReceiver; with dummy FE EoE data
10110101101100000100101111000000111000100100000000

The next step is to then prepare the MCC for triggers by issuing the EnDataTake command and issue a trigger:

EnDataTake;
10110101110000000
LV1;
11101

The output event data would then be decoded and checks made to verify that the parsed hit matched the hit which was loaded into the receiver.

7. FE REGISTER TEST PROCEDURES

The basic strategy for verifying the correct operation of FE registers involves loading the shift register (SR), latching the data, re-loading the SR with different data, reading the latched data back into the SR and then clocking the data out of the SR.

7.1 FE GLOBAL REGISTER TEST

Example sequence used to verify the proper operation of the 202-bit global register of FE chip 0:

WriteReg; register = FEEN, data = 0x0001 (FIFO 0)
WriteReg; register = CNT, with CCNT=32, DCNT=202
WriteFE; FE command = **CLOCK GLOBAL** with 'correct data' and DO-MUX=15
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE GLOBAL**
WriteFE; FE command = **NULL**
WriteReg; register = CNT, with CCNT=32, DCNT=202
WriteFE; FE command = **CLOCK GLOBAL** with 'wrong data'
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **READ GLOBAL**
WriteFE; FE command = **NULL**
WriteReg; register = CNT, with CCNT=32, DCNT=202
WrRdFE; FE command = **CLOCK GLOBAL** with 'correct data', global register content is clocked through the MCC back to the ROD for comparison with expectation

The return data stream should consist of the 5-bit header (11101) at 40MHz, followed by the command and data fields from the FE chip at 5MHz (i.e. each bit repeated 8 times). Upon receipt of the header therefore, the first 32 X 8 data bits may be ignored and the next 202 X 8 bits should be compared with expectation. One bit in the Global register is read-only (MonLeak status which is the MSB of the MonLeak ADC field; bit 35) and should therefore be ignored during the comparison.

7.1 FE PIXEL REGISTER TEST

Each of the 14 control bits may be tested independently in a similar manner to the global latches. In this case though there is no basic 'READ'-type command to reload the latch contents back into the shift register. A compound command containing several FE commands and MCC commands must be contructed for the READ process alone. The following is a breakdown of how an individual pixel control latch (Hitbus enable), may be tested for an FE (#14) with 8 working column pairs. In this nomenclature, when FE command types are summed it is implied that two or more command register bits are simulataneously set within a single command:

WriteReg; register = FEEN, data = 0x4000 (FIFO 14)
WriteReg; register = CNT, with CCNT=32, DCNT=202
WriteFE; FE command = **CLOCK GLOBAL** with DO-MUX=11 and the 8 working column pairs selected
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE GLOBAL**
WriteFE; FE command = **NULL**
WriteReg; register = CNT, with CCNT=32, DCNT=320X8=2560
WriteFE; FE command = **CLOCK PIXEL** with e.g. stored random pattern of 1s and 0s to 320*8 5MHz bits in the FE command data field
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE HITBUS**
WriteFE; FE command = **NULL**
WriteReg; register = CNT, with CCNT=32, DCNT=2560
WriteFE; FE command = **CLOCK PIXEL** with e.g. null data i.e. different from previous CLOCK PIXEL
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **READ PIXEL**
WriteFE; FE command = **WRITE HITBUS+READ PIXEL**
WriteReg; register = CNT, with CCNT=32, DCNT=2
WriteFE; FE command = **CLOCK PIXEL+WRITE HITBUS+READ PIXEL** with two null data bits
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE HITBUS+READ PIXEL**
WriteFE; FE command = **READ PIXEL**
WriteFE; FE command = **NULL**
WriteReg; register = CNT, with CCNT=32, DCNT=2560
WrRdFE; FE command = **CLOCK PIXEL** with 2560 (e.g. null) bits. Return data stream would be compared with original random mask to verify the correct operation of the Hitbus enable control latches.

8. GENERAL SCAN CONSIDERATIONS

Having verified the correct operation of a module with repect to its basic communication processes, one would then perform more complex tests which involve scanning. The readout logic in the FE chip is structured in such a way that at any one moment in time, no more than 64 hits may be present for any one of the 9 pairs of columns. Since there are 320 pixels in a column pair (CP), a maximum of 1/5 of pixels may be calibrated at the same time. Furthermore there is a limit to the number of hits which can be processed at the whole-FE chip level as a consequence of the FE-data receiver FIFOs on the MCC which are 128 ‘hits’ deep. In this context a *hit* may also be an `end-of-event word' (EoE), which in the FE-MCC protocol is the means by which the FE informs the outside world that it has transmitted the last physical hit corresponding to a particular level-1 trigger. Normally during calibration one treats each CP on the FE in the same way. Therefore, scans performed at the module level usually involve the simultaneous calibration of 10 pixels within a CP at the same time, (that being the largest factor of 320 which when multiplied by 9 gives a total hit count less than 112). For each trigger issued to the MCC, an FE may optionally be issued up to 16 contiguous level-1 pulses, (which is why the number of hits should be less than *112* as opposed to 128). This is particularly useful for calibration; the issuance of only a single level-1 to the FE chip would result in a biased threshold measurement due to timewalk. Also the relative timing of the calibration strobe w.r.t. the level-1 trigger does not need to be particularly precise if there is an effective trigger window, which is e.g. 8 BCOs long.

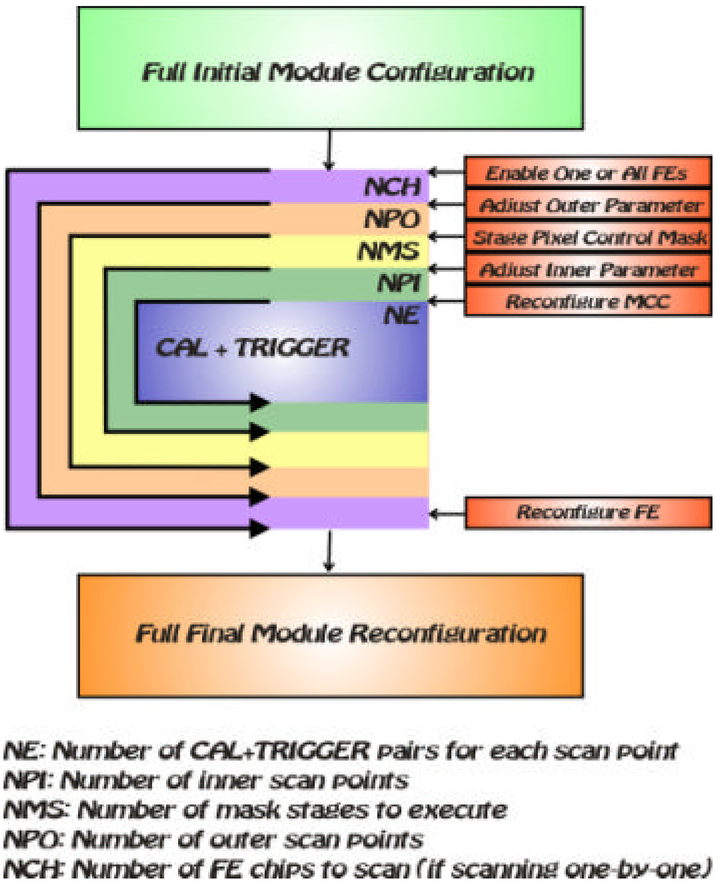
The usual methodology in covering the whole pixel array is to initially load a mask, (for the ENABLE and SELECT control bits), which selects a subset of pixels that are equally spaced according to the physical layout of the pixel register on the chip. This mask would then be shoved along through the array one step at a time. Meanwhile the calibration scan would be repeated at each step. An example of such a mask pattern might have only row 0 set in all of the even columns and row 159 set in all of the odd columns. This example would then have to be staged through the array 160 times to cover all of the pixels. In certain cases it makes sense to also stage the KILL control mask through the array at the same time with the inverse pattern, (i.e. in order to kill all pixels not currently being calibrated).

The scan process itself involves sending many pairs of CAL and TRIGGER fast commands to the MCC. The CAL command initiates the issuance of a calibration strobe (on a comman bus), to the FE chips. The properties of the strobe are set in the CAL and CNT registers prior to putting the MCC into trigger reception mode. The number of contiguous level-1 pulses to be issued to the FE chips, (upon receipt of a TRIGGER command), is set in the LV1 register. Before the MCC will respond to triggers the EnDataTake command must have been issued. When a slow command is received by the MCC it automatically exits take-data mode. The delays between the reception of CAL and TRIGGER by the MCC and the subsequent strobe and level-1 pulse issuances, are by necessity fixed. The time interval between the CAL and TRIGGER commands must also be constant (such that it properly corresponds to the trigger latency defined in the global registers on the FE chips), and adjustable over the whole FE latency range (0-255 BCOs). This responsibility resides at the ROD-crate level. At each scan point the number of CAL-TRIGGER pairs should be adjustable (to 16 bits).

In scanning modules there should exist the option of scanning every FE chip in tandem (concurrent-mode), or individually scanning one FE at a time.

9. SCAN PROCESS ANATOMY

At the most macroscopic level a calibration scan is perfromed in three stages: Configure the module, perform the scan then reconfigure the module. In the graphic the basic structure of a generic 2D scan is depicted. The scan portion itself comprises five nested loops. In the innermost loop (event loop) the pair of MCC fast commands which generate the calibration strobe and level-1 trigger, are just repeatedly sent NE times. The time interval between these pairs of commands must be sufficient to allow all of the upstream data to make it out of the module to avoid trigger pileup. This interval parameter should be adjustable to a full scale of 2^{13} units, a unit being 400ns. Before entering the event loop the MCC should be reconditioned in order to receive triggers and generate event data. This involves first writing to some registers to condition the trigger and level-1, then issuing the take data command. Here is an example of how the MCC should be reconditioned for event data:


Full Initial Module Configuration
NCH
NPO
NMS
NPI
NE
CAL + TRIGGER
Enable One or All FEs
Adjust Outer Parameter
Stage Pixel Control Mask
Adjust Inner Parameter
Reconfigure MCC
Reconfigure FE
Full Final Module Reconfiguration
NE: Number of CAL+TRIGGER pairs for each scan point
NPI: Number of inner scan points
NMS: Number of mask stages to execute
NPO: Number of outer scan points
NCH: Number of FE chips to scan (if scanning one-by-one)

WriteReg; register = CSR, with desired bandwidth, error-checking and ToT options
WriteReg; register = FEEN, with each of the 16 bits representing a particular receiver
WriteReg; register = CNT, to select the required strobe width
WriteReg; register = CAL, to enable the cal strobe and set the fine-delay
WriteReg; register = LV1, select the number of contiguous FE level-1 pulses per trigger
EnDataTake; render the module receptive to triggers

The FE chips must also be prepared for level-1 triggers by having the DO-MUX field in the global register set to 8. Since the only time an alternative would be used is for register readback tests, it's good practice to always load the value 8 whenever anything other than register tests are being performed, (e.g. during the full configuration at the start of scan). That way the global registers don't need to be reloaded as a matter of course at the MCC reconditioning stage

The next loop level in the 2D scan is the inner scan parameter incrementation, (for NPI scan points). In a typical threshold scan for example, the parameter would likely be the value of the VCAL DAC on the FE chip(s). The VCAL DAC (in FE-I1) is 9 bits and is used to determine the voltage which the internal choppers in each pixel chop down to (from the analogue supply voltage), in order to generate the voltage step which is applied to the calibration capacitors. A typical threshold scan choice might be to scan VCAL from 0 to 200 in NPI=201 steps for NE=100 events per point. In the pixel function library it makes sense to define a single function which contains knowledge of all of the possible parameters which one might want to scan. This would be called at each increment with the parameter identifier and the new value. If the scan parameter is a front-end variable, (as is usually the case), then this function would take care of relaying the new scan point data to all of the selected FE chips, in the case of concurrent-mode scanning.

The inner scan is repeated for each mask stage in order to cover all of the pixels in the FE arrays NMS times. The stage mask pattern would initially be loaded into the FE chips and then stepped through the array one shift-register location at a time. This works fine if only the patterns for the ENABLE and SELECT control bits are being staged. In FE-I1 a complication arises if the KILL mask needs to be staged also, (to keep untuned low-threshold or badly behaved pixels quiet). The pattern which should be staged for KILL has the opposite polarity to the others. Therefore one would need to either reload the entire shift register at each mask stage, (performing the stage process on the ROD), or utilise the READ PIXEL process in order to retrieve the latch contents for each control-bit before stepping the pattern along.

The next loop layer provides the option of performing a two-dimensional scan. An example of such an application is in the evaluation of timewalk in which the fine delay of the strobe pulse relative to the trigger (provided by the MCC), would be varied as the innermost scan parameter, while the magnitude of the injected charge would be scanned at the outer level. Finally in the case where for whatever reason the desire is to scan the module one FE at a time there is an outermost loop level to simply loop over the FEs. As the scan for each FE chip is completed it is restored to its normal configuration.

10. INITIAL CONFIGURATION PROCESS

At the initial configuration stage all of the FE and MCC register-based parameters are loaded. Also any requisite reset-like signals are issued, e.g. to synchronise the FEs within the module. For each pixel control latch (in each FE), a *static* mask is defined which corresponds to the 'normal' state for those latches (i.e. the state used during physics). For a 'good' pixel the states might be ENABLE=1, SELECT=0, KILL=0 and HITBUS=0. While a scan is in progress these settings are superceded by the staged mask for the particular FE (or FEs) which are currently being scanned. The unscanned chips remain in their normal static configuration. At the conclusion of the scan for each FE the static masks are reinstalled for that FE.

.The following is an example of the full configuration procedure for an individual FE (#7) with 9 working column pairs. The GAs in each FE command would be set to 7:

Reset the MCC:
ResetMCC;

Issue an FE hard reset:
ResetFE; with SyncW = 10. Applies to all FEs simultaneously

Load the global register:
WriteReg; register = FEEN, data = 0x0080 (FE 7)
WriteReg; register = CNT, with CCNT=32, DCNT=202
WriteFE; FE command = **CLOCK GLOBAL** with DO-MUX=8
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE GLOBAL**

Load the 5 TDAC control latches:
WriteReg; register = FEEN, data = 0x0080 (FE 7)
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the TDAC bit-0 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE TDAC0**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the TDAC bit-1 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE TDAC1**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the TDAC bit-2 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE TDAC2**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the TDAC bit-3 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE TDAC3**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the TDAC bit-4 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE TDAC4**

Load the 5 FDAC control latches:
WriteReg; register = FEEN, data = 0x0080 (FE 7)
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the FDAC bit-0 ON

WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE FDAC0**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the FDAC bit-1 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE FDAC1**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the FDAC bit-2 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE FDAC2**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the FDAC bit-3 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE FDAC3**
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the FDAC bit-4 ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE FDAC4**

Load the SELECT latches:
WriteReg; register = FEEN, data = 0x0080 (FE 7)
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the calibration select control bit ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE SELECT**

Load the ENABLE latches:
WriteReg; register = FEEN, data = 0x0080 (FE 7)
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the readout enable control bit ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE MASK**

Load the HITBUS latches:
WriteReg; register = FEEN, data = 0x0080 (FE 7)
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the hitbus participation control bit ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE HITBUS**

Load the KILL latches:
WriteReg; register = FEEN, data = 0x0080 (FE 7)
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with mask pattern corresponding to required selection of pixels which have the preamplifier disable control bit ON
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE KILL**

Reset the module BCO counter:
BCR;

Reset the module event counter:
ECR;

Simultaneously reset the 16 FE BCO counters:
ResetFE; with SyncW = 5

Simultaneously reset the 16 FE level-1 counters:
ResetFE; with SyncW = 2

11. SCAN MACHINERY DETAILS

Having reconditioned the MCC to accept triggers, the following pair of MCC fast commands are sent to the module with a precise delay in between in order to generate the calibration strobes and triggers. The interval in between adjacent pairs of fast commands should also be precisely counted:

CAL;
TRIGGER;

The process to load the staging mask into an FE (#8) with 9 working CPs will look something like the following:

WriteReg; register = FEEN, with 0x0100
WriteReg; register = CNT, with CCNT=32, DCNT=2880
WriteFE; FE command = **CLOCK PIXEL** with defined stage mask pattern in data field
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE MASK** to load the initial pattern into the readout-enable latches
WriteFE; FE command = **WRITE SELECT** to load the initial pattern into the cal select latches

Then at each subsequent mask stage the following commands would effect a stepping of the pattern by one shift register location. The sense of the bit entered into the CLOCK

PIXEL command should be the same as the bit which effectively drops off the end of the register, in order to facilitate wrap-around. This way the initial pattern will not need to be re-loaded at the beginning of each full mask stage process:

WriteReg; register = FEEN, with 0x0100
WriteReg; register = CNT, with CCNT=32, DCNT=1
WriteFE; FE command = **CLOCK PIXEL** with a single-bit (at 5MHz) data field
WriteReg; register = CNT, with CCNT=32, DCNT=0
WriteFE; FE command = **WRITE MASK** to load the initial pattern into the readout-enable latches
WriteFE; FE command = **WRITE SELECT** to load the initial pattern into the cal select latches

12. UPSTREAM MODULE EVENT DATA FORMAT

Each pixel module should normally respond to trigger commands with recognisable serial return data streams. In very exceptional circumstances, an MCC may fail to respond. An example of such a case may arise because a receiver channel, which is enabled in the FEEN register, does not see any data corresponding to a transmitted level-1 trigger. This could theoretically occur if the DO-MUX global-register field in an FE chip was corrupted by an SEU, thus preventing the correct data source from being MUXed to its output (DO) pin. The MCC event-building algorithm will wait *ad infinitum* until it sees recognisable data from all enabled FEs, before transmitting event packets on its DTO pin(s). In the case that a module ceases to respond to TRIGGER commands, the module should be completely reset and reconfigured. If this were to occur during calibration, the whole scan would need to be regenerated from scratch.

This is the MCC to ROD event data protocol for MCC-I1; extracted from [ref]:

<Event>
::= <Header> L1ID <S> BCID <MccFlag>? <FrontEnd>* <Trailer>

<Header>
::= 11101

<S>
::= 1

<MccFlag>
::= <Sync> MCC-FLAG

<Sync>
::= <S>
||= <<S> NULL>+

<FrontEnd>
::= <Sync> MCC-FE# <Hit>+ <FeFlag>?
||= <Sync> MCC-FE# <Hit>* <FeFlag>

<Hit>
::= <Sync> ROW# COL#
||= <Sync> ROW# COL# TOT if ToT option selected in CSR)

<FeFlag>
::= <Sync> FE-FLAG

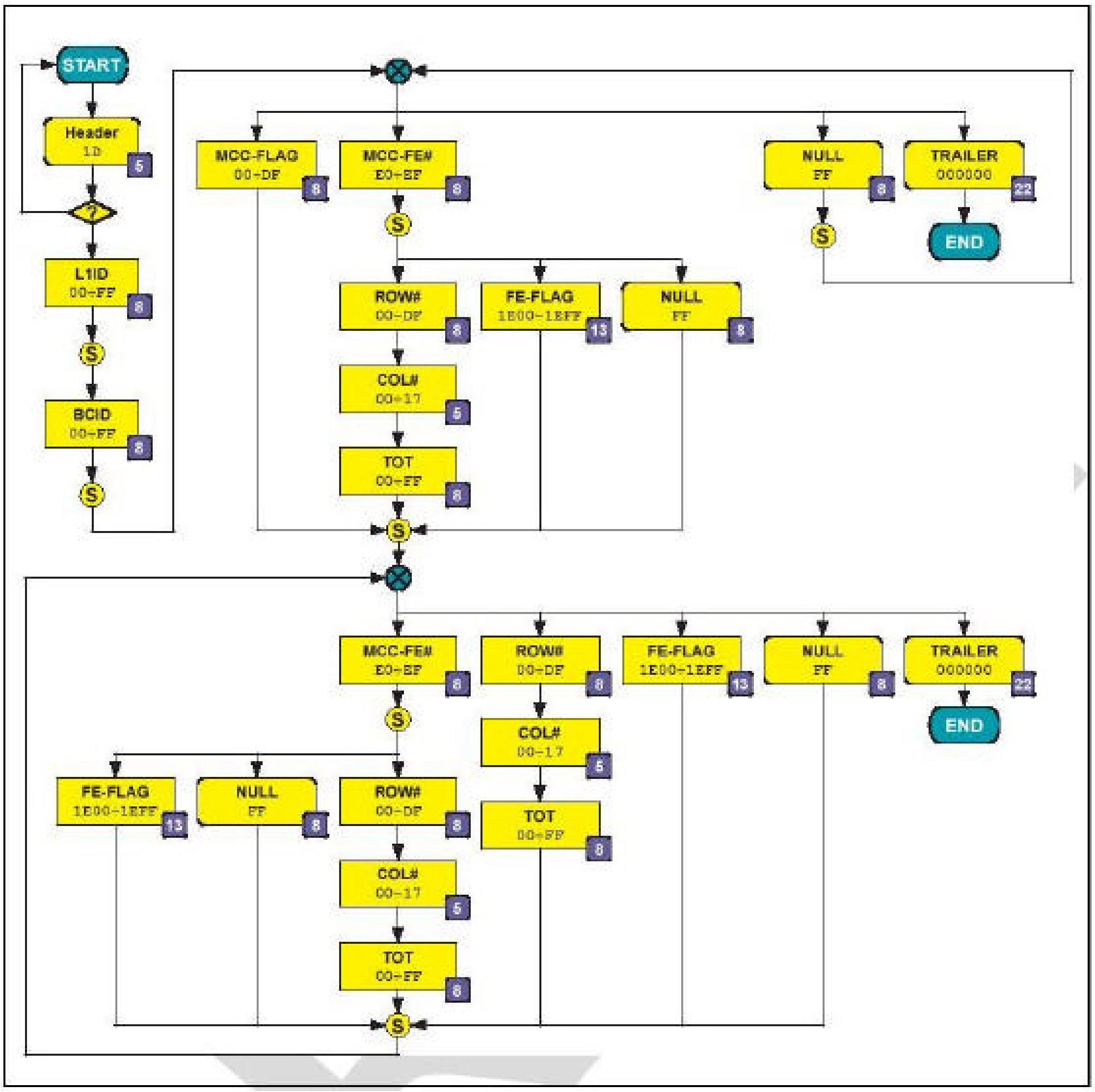
<Trailer>
::= <S> 00 0000 0000 0000
||= <S> 00 0000 0000 0000 0000 0000 (if ToT option selected in CSR)

where
::= is a definition, ||= is an alternative definition, upper case words are defined in the keyword table (below), ? is an optional field, * = 0,1 or more items and + is 1 or more items.

Keyword table:

Keyword	**Minimum**	**Maximum**	**Description**
BCID	0000 0000	1111 1111	Bunch Crossing ID
COL#	0 0000	10111	Column number
FE-FLAG	1 1110 FFFF MMMM		Error/Warning F=FE, M=MCC
LVID	0000 0000	1111 1111	Level-1 trigger ID
MCC-FE#	1110 0000	1110 1111	FE number according to MCC receiver/FEEN
MCC-FLAG	0000 0000	1101 1111	Error/Warning from MCC
NULL	1111 1111		Null data
ROW#	0000 0000	1101 1111	Pixel row number
TOT	0000 0000	1111 1111	Time-over-threshold for the hit

The following flowchart also taken from [ref] describes an appropriate MCC data decoding algorithm:



12. CALIBRATION CASE 1: SIMPLE 1D THRESHOLD SCAN

The object of running a simple 1D threshold scan is to determine the noise and threshold of each pixel within the module. This would be performed right after a (more complex) TDAC-tune scan in order to evaluate the quality of the threshold matching. It would also be the means by which the threshold dispersion and noise were necessarily, periodically monitored in the experiment. The evidence of radiation tolerance evaluations performed with FE-I1, suggests that the threshold dispersion can be expected to increase as a function of ionising dose. The noise is naturally expected to grow slowly with dose, due to enhancements of the sensor leakage-current and interpixel-capacitance.

The strategy consists of making multiple calibration-charge injections into each preamplifier whilst scanning the magnitude of the charge, (i.e. the VCAL DAC setting), from e.g. zero, up to a value which is comfortably above the level of highest expected individual discriminator threshold. For each pixel a histogram is developed showing the number of hits returned for each charge setting which, under normal circumstances, would be 0% well below threshold and 100% above. The step between the two levels has a finite width due to the noise, which may be thought of as a statistical (Gaussian) variation in the observed threshold. In fact the variation mostly arises from the uncertainty in the height of the pulse as observed by the discriminator. This gives the histogram an *s-curve* profile. The step width is used to evaluate the noise, so it's important to ensure that the step-size chosen for the scan is sufficiently fine to obtain a good fit within the region of the s-curve. In FE-I1 the calibration of the VCAL DAC is ~45e- per count when the low value injection capacitor is in use. The lowest noise one might expect to encounter, (arising from the case that a pixel has a missing bump, thus practically no load capacitance), is around 140e-. Therefore, a single DAC count should be set for the granularity of the scan.